

"Coding is Rhyme"

Handy Tricks For Beginners & Professionals

SQL Server 2012, 2008 - Tutorial, Article, Archive, Pdf, Handy Tricks, Training, Code Snippets, Reference Manual, T-SQL, PL/SQL, Development, Administration, Performance Tuning, Backup, Recovery, Reporting, Analysis, Security, XML, CLR Objects

Custom Servers Solutions

 softlayer.com/dedicated-server

Secure, Reliable Dedicated Servers 24/7 Customer Support. 0% Downtime.

Different Types of SQL Joins

Posted By : Shailendra Chauhan, 02 Mar 2011

Updated On : 07 Dec 2012

Version Support : SQL Server 2005,2008,2012

Keywords : *Types of Sql Join pdf, Introduction to Sql Join, Inner Join, Outer Join, Left Outer Join, Right Outer Join, Full Outer Join, Cross Join, Self Join*

Sql joins are used to fetch/retrieve data from two or more data tables, based on a join condition. A join condition is a relationship among some columns in the data tables that take part in Sql join. Basically data tables are related to each other with keys. We use these keys relationship in sql joins. Also, refer the article [SQL Joins with C# LINQ](#).

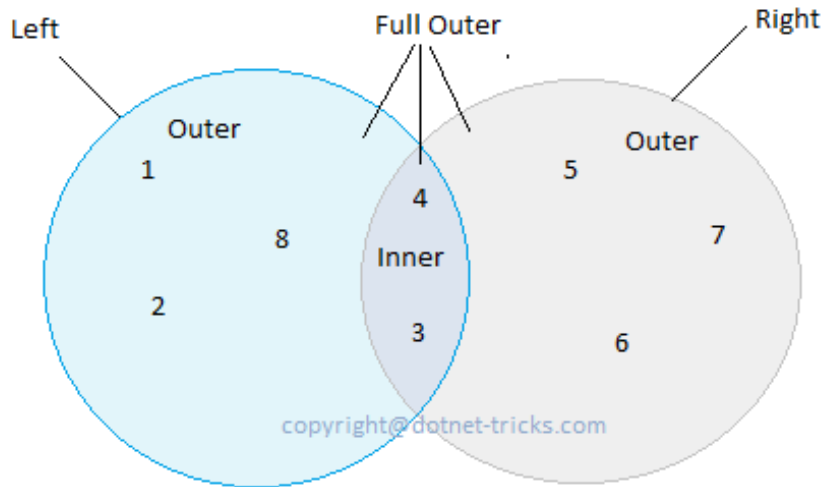
Types of Joins

In Sql Server we have only three types of joins. Using these joins we fetch the data from multiple tables based on condition.

01. Inner Join

Inner join returns only those records/rows that match/exists in both the tables. Syntax for Inner Join is as

```
1.  Select * from table_1 as t1
2.  inner join table_2 as t2
3.  on t1.IDcol=t2.IDcol
```

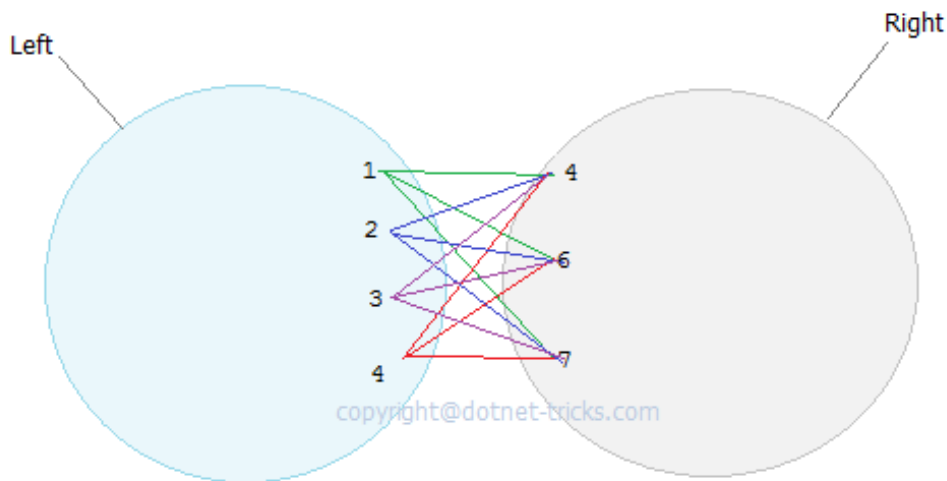


Inner Join Result : (4,3)

Left Join Result : (1,2,8,4,3)

Right Join Result : (5,6,7,4,3)

Full Outer Join Result : (1,2,8,4,3,5,6,7)



Cross Join Result : ((1,4), (1,6), (1,7), (2,4), (2,6), (2,7), (3,4), (3,6), (3,7), (4,4), (4,6), (4,7))

02. Outer Join

We have three types of Outer Join.

01. Left Outer Join

Left outer join returns all records/rows from left table and from right table returns only matched records. If there are no columns matching in the right table, it returns NULL values. Syntax for Left outer Join is as :

```
1. Select * from table_1 as t1
```

```
2. left outer join table_2 as t2
3. on t1.IDcol=t2.IDcol
```

02. Right Outer Join

Right outer join returns all records/rows from right table and from left table returns only matched records. If there are no columns matching in the left table, it returns NULL values. Syntax for right outer Join is as :

```
1. Select * from table_1 as t1
2. right outer join table_2 as t2
3. on t1.IDcol=t2.IDcol
```

03. Full Outer Join

Full outer join combines left outer join and right outer join. This join returns all records/rows from both the tables. If there are no columns matching in the both tables, it returns NULL values. Syntax for full outer Join is as :

```
1. Select * from table_1 as t1
2. full outer join table_2 as t2
3. on t1.IDcol=t2.IDcol
```

03. Cross Join

Cross join is a cartesian join means cartesian product of both the tables. This join does not need any condition to join two tables. This join returns records/rows that are multiplication of record number from both the tables means each row on left table will related to each row of right table. Syntax for right outer Join is as :

```
1. Select * from table_1
2. cross join table_2
```

04. Self Join

Self join is used to join a database table to itself, particularly when the table has a Foreign key that references its own Primary Key. Basically we have only three types of joins : Inner join, Outer join and Cross join. We use any of these three JOINS to join a table to itself. Hence Self join is not a type of Sql join.

Join Examples

Suppose we following three tables and data in these three tables is shown in figure. You can download the SQL script used in this article by using [link](#).

	CustID	Name	Address	ContactNo	
1	1	Sam	New Delhi	9555555555	tblCustomer
2	2	Rahul	Gurgaon	9666666666	
3	3	Hans	Noida	9444444444	
4	4	Jeetu	Delhi	9333333333	
5	5	Ankit	Noida	9222222222	

	ProductID	Name	UnitPrice	CatID	EntryDate	ExpiryDate
1	1	Dell Computer	25000	1	2012-10-16 23:05:05.550	2012-10-16 23:05:05.550
2	2	HCL Computer	20000	1	2012-10-16 23:05:46.990	2012-10-16 23:05:46.990
3	3	Apple Mobile	40000	3	2012-10-16 23:06:11.283	2012-10-16 23:06:11.283
4	4	Samsung Mobile	25000	3	2012-10-16 23:06:28.727	2012-10-16 23:06:28.727
5	5	Sony Laptop	35000	2	2012-10-16 23:06:52.143	2012-10-16 23:06:52.143
6	6	Dell Laptop	36000	2	2012-10-16 23:07:07.380	2012-10-16 23:07:07.380
7	7	HP Printer	12000	4	2012-10-16 23:07:35.010	2012-10-16 23:07:35.010
8	8	Canon Printer	10000	4	2012-10-16 23:07:54.213	2012-10-16 23:07:54.213

	OrderID	ProductID	Quantity	Price	CustomerID	ContactNo
1	1	1	6	150000	1	9555555555
2	2	2	4	80000	2	NULL
3	3	2	2	40000	3	9444444444
4	4	3	5	200000	4	9333333333
5	5	5	1	35000	5	9666666666

	OrderID	ProductID	Quantity	Price	CustomerID	ContactNo
1	1	1	6	150000	1	9555555555
2	2	2	4	80000	2	NULL
3	3	2	2	40000	3	9444444444
4	4	3	5	200000	4	9333333333
5	5	5	1	35000	5	9666666666

	CustID	Name	Address	ContactNo	
1	1	Sam	New Delhi	9555555555	tblCustomer
2	2	Rahul	Gurgaon	9666666666	
3	3	Hans	Noida	9444444444	
4	4	Jeetu	Delhi	9333333333	
5	5	Ankit	Noida	9222222222	

	ProductID	Name	UnitPrice	CatID	EntryDate	ExpiryDate
1	1	Dell Computer	25000	1	2012-10-16 23:05:05.550	2012-10-16 23:05:05.550
2	2	HCL Computer	20000	1	2012-10-16 23:05:46.990	2012-10-16 23:05:46.990
3	3	Apple Mobile	40000	3	2012-10-16 23:06:11.283	2012-10-16 23:06:11.283
4	4	Samsung Mobile	25000	3	2012-10-16 23:06:28.727	2012-10-16 23:06:28.727
5	5	Sony Laptop	35000	2	2012-10-16 23:06:52.143	2012-10-16 23:06:52.143
6	6	Dell Laptop	36000	2	2012-10-16 23:07:07.380	2012-10-16 23:07:07.380
7	7	HP Printer	12000	4	2012-10-16 23:07:35.010	2012-10-16 23:07:35.010
8	8	Canon Printer	10000	4	2012-10-16 23:07:54.213	2012-10-16 23:07:54.213

	OrderID	ProductID	Quantity	Price	CustomerID	ContactNo
1	1	1	6	150000	1	9555555555
2	2	2	4	80000	2	NULL
3	3	2	2	40000	3	9444444444
4	4	3	5	200000	4	9333333333
5	5	5	1	35000	5	9666666666

	OrderID	ProductID	Quantity	Price	CustomerID	ContactNo
1	1	1	6	150000	1	9555555555
2	2	2	4	80000	2	NULL
3	3	2	2	40000	3	9444444444
4	4	3	5	200000	4	9333333333
5	5	5	1	35000	5	9666666666

Inner Join

1. **SELECT** t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price

2. **FROM** tblProduct **AS** t0
3. **INNER JOIN** tblOrder **AS** t1 **ON** t0.ProductID = t1.ProductID
4. **ORDER BY** t1.OrderID

Results

Messages

	OrderID	ProductID	Name	UnitPrice	Quantity	Price
1	1	1	Dell Computer	25000	6	150000
2	2	2	HCL Computer	20000	4	80000
3	3	2	HCL Computer	20000	2	40000
4	4	3	Apple Mobile	40000	5	200000
5	5	5	Sony Laptop	35000	1	35000

Inner Join among more than two tables

1. **SELECT** t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price, t2.Name **AS** Customer
2. **FROM** tblProduct **AS** t0
3. **INNER JOIN** tblOrder **AS** t1 **ON** t0.ProductID = t1.ProductID
4. **INNER JOIN** tblCustomer **AS** t2 **ON** t1.CustomerID = t2.CustID
5. **ORDER BY** t1.OrderID

Results

Messages

	OrderID	ProductID	Name	UnitPrice	Quantity	Price	Customer
1	1	1	Dell Computer	25000	6	150000	Sam
2	2	2	HCL Computer	20000	4	80000	Rahul
3	3	2	HCL Computer	20000	2	40000	Hans
4	4	3	Apple Mobile	40000	5	200000	Jeetu
5	5	5	Sony Laptop	35000	1	35000	Ankit

Inner Join on multiple conditions

1. **SELECT** t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price, t2.Name **AS** Customer
2. **FROM** tblProduct **AS** t0
3. **INNER JOIN** tblOrder **AS** t1 **ON** t0.ProductID = t1.ProductID
4. **INNER JOIN** tblCustomer **AS** t2 **ON** t1.CustomerID = t2.CustID **AND** t1.ContactNo = t2.ContactNo
5. **ORDER BY** t1.OrderID

Results Messages

	OrderID	ProductID	Name	UnitPrice	Quantity	Price	Customer
1	1	1	Dell Computer	25000	6	150000	Sam
2	3	2	HCL Computer	20000	2	40000	Hans
3	4	3	Apple Mobile	40000	5	200000	Jeetu

Left Outer Join

1. **SELECT** t1.OrderID **AS** OrderID , t0.ProductID , t0.Name , t0.UnitPrice ,
t1.Quantity **AS** Quantity , t1.Price **AS** Price
2. **FROM** tblProduct **AS** t0
3. **LEFT OUTER JOIN** tblOrder **AS** t1 **ON** t0.ProductID = t1.ProductID
4. **ORDER BY** t0.ProductID
- 5.

	OrderID	ProductID	Name	UnitPrice	Quantity	Price
1	1	1	Dell Computer	25000	6	150000
2	2	2	HCL Computer	20000	4	80000
3	3	2	HCL Computer	20000	2	40000
4	4	3	Apple Mobile	40000	5	200000
5	NULL	4	Samsung Mobile	25000	NULL	NULL
6	5	5	Sony Laptop	35000	1	35000
7	NULL	6	Dell Laptop	36000	NULL	NULL
8	NULL	7	HP Printer	12000	NULL	NULL
9	NULL	8	Canon Printer	10000	NULL	NULL

Right Outer Join

1. **SELECT** t1.OrderID **AS** OrderID , t0.ProductID , t0.Name , t0.UnitPrice ,
t1.Quantity **AS** Quantity , t1.Price **AS** Price
2. **FROM** tblProduct **AS** t0
3. **RIGHT OUTER JOIN** tblOrder **AS** t1 **ON** t0.ProductID = t1.ProductID
4. **ORDER BY** t0.ProductID

	OrderID	ProductID	Name	UnitPrice	Quantity	Price
1	1	1	Dell Computer	25000	6	150000
2	2	2	HCL Computer	20000	4	80000
3	3	2	HCL Computer	20000	2	40000
4	4	3	Apple Mobile	40000	5	200000
5	5	5	Sony Laptop	35000	1	35000

Full Outer Join

1. **SELECT** t1.OrderID **AS** OrderID , t0.ProductID , t0.Name , t0.UnitPrice ,
t1.Quantity **AS** Quantity , t1.Price **AS** Price
2. **FROM** tblProduct **AS** t0
3. **FULL OUTER JOIN** tblOrder **AS** t1 **ON** t0.ProductID = t1.ProductID
4. **ORDER BY** t0.ProductID

	OrderID	ProductID	Name	UnitPrice	Quantity	Price
1	1	1	Dell Computer	25000	6	150000
2	2	2	HCL Computer	20000	4	80000
3	3	2	HCL Computer	20000	2	40000
4	4	3	Apple Mobile	40000	5	200000
5	NULL	4	Samsung Mobile	25000	NULL	NULL
6	5	5	Sony Laptop	35000	1	35000
7	NULL	6	Dell Laptop	36000	NULL	NULL
8	NULL	7	HP Printer	12000	NULL	NULL
9	NULL	8	Canon Printer	10000	NULL	NULL

Cross Join

1. **SELECT** t1.OrderID, t0.ProductID, t0.Name, t0.UnitPrice, t1.Quantity, t1.Price
2. **FROM** tblProduct **AS** t0, tblOrder **AS** t1
3. **ORDER BY** t0.ProductID

	OrderID	ProductID	Name	UnitPrice	Quantity	Price
1	1	1	Dell Computer	25000	6	150000
2	2	1	Dell Computer	25000	4	80000
3	3	1	Dell Computer	25000	2	40000
4	4	1	Dell Computer	25000	5	200000
5	5	1	Dell Computer	25000	1	35000
6	1	2	HCL Computer	20000	6	150000
7	2	2	HCL Computer	20000	4	80000
8	3	2	HCL Computer	20000	2	40000
9	4	2	HCL Computer	20000	5	200000
10	5	2	HCL Computer	20000	1	35000
11	1	3	Apple Mobile	40000	6	150000
12	2	3	Apple Mobile	40000	4	80000
13	3	3	Apple Mobile	40000	2	40000
14	4	3	Apple Mobile	40000	5	200000
15	5	3	Apple Mobile	40000	1	35000
16	1	4	Samsung Mobile	25000	6	150000
17	2	4	Samsung Mobile	25000	4	80000
18	3	4	Samsung Mobile	25000	2	40000
19	4	4	Samsung Mobile	25000	5	200000
20	5	4	Samsung Mobile	25000	1	35000
21	1	5	Sony Laptop	35000	6	150000
22	2	5	Sony Laptop	35000	4	80000
23	3	5	Sony Laptop	35000	2	40000
24	4	5	Sony Laptop	35000	5	200000
25	5	5	Sony Laptop	35000	1	35000
26	1	6	Dell Laptop	36000	6	150000
27	2	6	Dell Laptop	36000	4	80000
28	3	6	Dell Laptop	36000	2	40000
29	4	6	Dell Laptop	36000	5	200000
30	5	6	Dell Laptop	36000	1	35000
31	1	7	HP Printer	12000	6	150000
32	2	7	HP Printer	12000	4	80000
33	3	7	HP Printer	12000	2	40000
34	4	7	HP Printer	12000	5	200000
35	5	7	HP Printer	12000	1	35000
36	1	8	Canon Printer	10000	6	150000
37	2	8	Canon Printer	10000	4	80000
38	3	8	Canon Printer	10000	2	40000
39	4	8	Canon Printer	10000	5	200000
40	5	8	Canon Printer	10000	1	35000

Self Join

To understand Self Join, suppose we following two tables and data in these two tables is shown in figure.

1. **CREATE TABLE emp**


```

2. (
3. id int NOT NULL primary key,
4. name varchar(100) NULL,
5. designation varchar(50) NULL,
6. supid int foreign key references emp(id) ) -- In this table we have a Foreign key
    supid that references its own Primary Key id. We use it for Self Join
7. INSERT INTO emp(id,name,designation) VALUES(1,'mohan','Manger')
8. INSERT INTO emp(id,name,designation,supid) VALUES(2,'raj kumar','SE',1)
9. INSERT INTO emp(id,name,designation) VALUES(3,'bipul kumar','Manager')
10. INSERT INTO emp(id,name,designation,supid) VALUES(4,'mrinal kumar','SE',2)
11. INSERT INTO emp(id,name,designation,supid) VALUES(5,'jitendra kumar','SE',2)

```

Results		Messages		
	id	name	designation	supid
1	1	mohan	Manger	NULL
2	2	raj kumar	SE	1
3	3	bipul kumar	Manager	NULL
4	4	mrinal kumar	SE	2
5	5	jitendra kumar	SE	2

```

1. CREATE TABLE empinfo
2. (
3. id int primary key,
4. address varchar(50) NULL
5. )
6. INSERT INTO empinfo(id,address) VALUES(1,'Delhi')
7. INSERT INTO empinfo(id,address) VALUES(2,'Noida')
8. INSERT INTO empinfo(id,address) VALUES(4,'Gurgaon')
9. INSERT INTO empinfo(id,address) VALUES(6,'Delhi')
10. INSERT INTO empinfo(id,address) VALUES(7,'Noida')

```

Results		Messages	
	id	address	
1	1	Delhi	
2	2	Noida	
3	4	Gurgaon	
4	6	Delhi	
5	7	Noida	

```

1. select e.id,e.name,e.supid as managerid, ei.name as managername from emp e left
    join emp ei on e.supid=ei.id;
2. -- outer keyword is optional

```

Results		Messages		
	id	name	managerid	managemame
1	1	mohan	NULL	NULL
2	2	raj kumar	1	mohan
3	3	bipul kumar	NULL	NULL
4	4	mrinal kumar	2	raj kumar
5	5	jitendra kumar	2	raj kumar

What do you think?

I hope you will enjoy these valuable tricks while query the data from database like SQL Server. I would like to have feedback from my blog readers. Your valuable feedback, question, or comments about this article are always welcome.

Share this article with your friends!

Share { 9 }

Tweet { 0 }

G+1 { 11 }

Share 4

About the Author



Shailendra Chauhan works as Software Analyst at reputed MNC and has more than 5 years of hand over Microsoft .NET technologies. He is a .NET Consultant and is the founder & chief editor of www.dotnet-tricks.com and www.dotnetinterviewtricks.com blogs. He is an author of book [ASP.NET MVC Interview Questions and Answers](#).

He loves to work with web applications and mobile apps using Microsoft technology including ASP.NET, MVC, C#, SQL Server, WCF, Web API, Entity Framework, Cloud Computing, Windows Azure, jQuery, jQuery Mobile, Knockout.js, Angular.js and many more web technologies. [More...](#)

« Prev

Next »

Recommended Articles!

1. Introduction to SQL Server

Dot Net Tricks

4. [Insert, Update, Delete Operations using Stored Procedures](#)
5. [Group By and Having Clause](#)
6. [Views](#)
7. [Triggers](#)
8. [Index Views](#)
9. [Linking](#)
10. [Tables or Constraints](#)

24 [CLICK HERE to book an appointment](#) [Dot Net Tricks](#)

[Login](#) ▾

So [Dr Batra's](#) [Share](#) [Favorite](#) ★

[Join the discussion...](#)

[farhan](#) • 17 days ago
very nice article
^ | ▾ • [Reply](#) • [Share](#) ›

[Zikriya Hafizi](#) • a month ago
Much sophisticated site for beginners as well as advance students! I like it very much.
^ | ▾ • [Reply](#) • [Share](#) ›

Join our Training programs in **Delhi/Noida** on
ASP.NET MVC, WCF, EF, jQuery - call us **+91-**

9871749695 [Reply](#) • [Share](#) ›



Bhagyashree • 3 months ago

thanks a lot!!!....It really helps :)

^ | v • [Reply](#) • [Share](#) ›



Muhammad Rasheed • 3 months ago

Thanks sir...It is very useful , simple and easy to understand

^ | v • [Reply](#) • [Share](#) ›



Sajjad Aslam • 4 months ago

I personally likes your site content

^ | v • [Reply](#) • [Share](#) ›



tyt@gmail.com • 8 months ago

tytytytyty

^ | v • [Reply](#) • [Share](#) ›



tyt@gmail.com • 8 months ago

rtrt

^ | v • [Reply](#) • [Share](#) ›



Rajagopal • 8 months ago

Excellent Article..

^ | v • [Reply](#) • [Share](#) ›



harika • 9 months ago

very clear presentation.. by seeing your examples we can understand

^ | v • [Reply](#) • [Share](#) ›



kanna • 10 months ago

fine..

^ | v • [Reply](#) • [Share](#) ›



ankur • 11 months ago

It is nice . You can further improve it in examples by first mentioning the requirement & then writing the SQL. Thanks.

^ | v • [Reply](#) • [Share](#) ›



sreenu • a year ago

First of all congrats to you ...you did a grate job...really excellent examples .every one can easily learn about sql joins throw this example. ...

^ | v • [Reply](#) • [Share](#) ›



avinash • a year ago

Very Nice.



1 ^ | v • Reply • Share ›



Neelesh Gupta • a year ago

Nice explanation !

^ | v • Reply • Share ›



Hamid • a year ago

Good one

^ | v • Reply • Share ›



kedar • a year ago

Easy to understand...thanks!

^ | v • Reply • Share ›



balasubramaniyan • a year ago

nice article..... good job.....

^ | v • Reply • Share ›



Ravindra Pratap • a year ago

great introduction sir ji

^ | v • Reply • Share ›



santhosh • a year ago

Very good article.Simple and easy to understand.

^ | v • Reply • Share ›



anurag • a year ago

i like your explanation and it is very useful as well as very easy to understand

^ | v • Reply • Share ›



Toshim Shaikh • a year ago

nice article ,it is really beneficial for me

^ | v • Reply • Share ›



shanmukh gona • a year ago

very good article, really good for all

^ | v • Reply • Share ›



Md Zameer Khan • 2 years ago

This is really a nice article Shailendra. Very precise. Kudos

^ | v • Reply • Share ›

ALSO ON DOT NET TRICKS

WHAT'S THIS?

Securing ASP.NET Web API using basic Authentication

CRUD Operations using jQuery dialog, Entity Framework and ASP.NET MVC

2 comments • 3 months ago



Pharylon — Very useful information! Thanks a ton!

ASP.NET MVC Interview Questions and Answers Book

19 comments • a month ago



Shailendra Chauhan — Now, I have updated the article with direct download link... :)

13 comments • 2 months ago



kumar — your Source Code is not working .. please help sir!

Differences between Object, Var and Dynamic type

3 comments • 3 months ago



ram — hhhhhh

[Subscribe](#)[Add Disqus to your site](#)

Search Articles

Subscribe & follow Us

PROTECTED BY **COPYSCAPE** DO NOT COPY

Free ASP.NET MVC Interview Book



Learn In Hindi



Browse By Category

.Net Framework 7 ADO.NET 6

Ajax 2 ASP.NET 15

ASP.NET Web Page 1 Backbone 1

C 5 C# 29

C# Windows Apps 6 CSS 5

Dependency Injection 4 Design Patterns 18

Entity Framework 10 Fluent Nhibernate 1

JavaScript 6 jQuery 4

jQuery Mobile 6 Knockout 5

LINQ 7 MVC 54

MVC 6 1 Nhibernate 1

OOPS 6 SQL Server 47

SQL Server 2012 1 SVN 1

TFS 3 Visual Studio 3

WCF 10 Web API 7

Web Service 2 Windows Azure 2

Windows Phone Apps 6 WPF 4

Recent Articles

- [Comparing LINQ with Stored Procedure](#)
- [Introduction to ASP.NET vNext and MVC 6](#)
- [Understanding Attribute Routing in ASP.NET MVC](#)
- [Understanding decision making statements in C#](#)
- [Difference between ref and out parameters](#)
- [Defining Namespace or Nested Objects in JavaScript](#)
- [Understanding Prototype in JavaScript](#)
- [Download SQL Server 2014 Express from Microsoft download center](#)
- [What is Backbone.js and why to use it?](#)
- [Difference between LINQ to SQL and Entity Framework](#)

Popular Articles

- [Detailed ASP.NET MVC Pipeline](#)
- [Custom Authentication and Authorization in MVC](#)
- [Understanding Entity Framework Code First Migrations](#)
- [Understanding Detailed Architecture of ASP.NET 4.5](#)
- [Gang of Four \(GOF\) Design Patterns in .Net](#)
- [Understanding Inversion of Control, DI and SL](#)
- [Difference between WCF and Web API and WCF REST](#)
- [CRUD Operations using jQuery dialog, EF and MVC](#)
- [SQL Joins with C# LINQ](#)
- [.Net Garbage Collection in depth](#)

Like us on Facebook

Featured AD

Who The?

If you're looking to learn a trick or technique which you have seen somewhere else. Let me know and I'll do my best to explain how it was done. If I don't know how to do, I'll take it as an exciting topic & I'll find upright resources and share. [More..](#)

Disclaimer!

This is my personal blog and having articles on .net and others .net related technologies. The opinions expressed here are my own and not belongs to my employer or other organization. I have listed my own learning experience on this blog.

Copyright © 2012-2014 The content is copyright to Shailendra Chauhan and may not be reproduced on other websites without permission from the owner.